INTERNATIONAL JOURNAL OF ENHANCED RESEARCH IN SCIENCE TECHNOLOGY & ENGINEERING
VOL. 2 ISSUE 3, MARCH-2013 ISSN NO: 2319-7463

Encoding of Low-Density Parity-Check codes for CCSDS Standard

Goutham S. Shetty[1], R. V. Nadagouda[2], Satheesh Rao[3]
[1]Dept. of Electronics and Communication, NMAM Institute of Technology, Nitte, Karnataka, India
[2]Scientist/Engineer 'SG' (Head, BCSS/CBRD/SCG-ISAC), ISRO Satellite Centre, Bangalore, India
[3]Asst. Professor Gd-3 Dept. of Electronics and Communication, NMAM Institute of Technology, Nitte, Karnataka, India
[1]gowthammetch2013@yahoo.com, [2]rvngowda@isac.gov.in, [3]write2sat@gmail.com

Abstract: Low-Density Parity-Check codes (LDPC) are found to be efficient at high code rates and have a low error probability and also provide a higher coding gain when compared to other coding techniques working in this domain. This paper proposes a LDPC encoder and also construction of parity check matrices for different code rates, as specified in the CCSDS 131.1-0-2 document [8]. To evaluate the encoder a VHDL description was developed and synthesized on an Altera Cyclone platform for the CCSDS standard.

Keywords: Low-Density Parity-Check codes (LDPC); Quasi-Cyclic LDPC; Attach Sync Marker (ASM); Consultative Committee for Space Data System (CCSDS).

Introduction

LDPC codes were originally introduced by Gallager [1] in the 1960s. However, due to the lack of an efficient decoding algorithm and hardware capabilities, the codes were not widely used at the time and slowly faded away. In the 1990s, LDPC codes were rediscovered and were shown to have performance close to the Shannon limit. LDPC codes have received a great deal of attention since the rediscovery of their outstanding potential. Along with the increase in computer processing power, the once considered impractical codes have long since proved to be quite usable in modern transmission systems. In 1997, Luby et al. considered LDPC codes to be extremely useful for applications such as real-time audio and video transmission over the Internet.

Low-density parity-check codes are a class of linear block code defined by a sparse M×N parity-check matrix, H [1], where N > M and M = N - K. Although LDPC codes can be generalized to non-binary symbols, we consider only binary codes. The parity-check matrix has a small number of '1' entries compared to '0' entries, making it sparse. The number of '1's in a parity-check matrix row is called the row-weight, k, and the number of '1's in a column is the column-weight, j. A regular LDPC code is one in which both row and column weights are constant; otherwise the parity check matrix is irregular. Row and column weights are much smaller than the matrix dimensions, with row weights greater than column weights. The rate of the parity check or code matrix is the fraction of information bits in the codeword. It is given by K/N = (N-M)/N = 1-(M/N). The number of '1' entries in the parity-check matrix is given by Mk or Nj. From Mk = Nj, we get M/N = j/k. Hence, the rate of matrix could also be expressed as 1-(j/k) [9].

One class of structured LDPC codes that allows low-complexity encoding is the class of quasi-cyclic (QC)-LDPC codes. It is known in coding theory [11], [12] that QC codes can be encoded with simple shift registers, with linear complexity based on their generators (or generator matrices). Well-designed QC-LDPC codes have been shown to perform as well as computer-generated random LDPC codes, regular or irregular [6], in terms of bit-error performance, block-error performance and error floor, collectively. Therefore, in practical applications, they are strong competitors to the random codes, due to their simple encoding and low error floors. These codes also have advantages in integrated circuit (IC) decoder implementations due to their cyclic symmetry, which results in simple regular wiring and modular structure.

Introduction to QC-LDPC codes

Quasi-Cyclic (QC) LDPC codes are codes in which rows or columns in a sub-matrix have similar and cyclic connections. Due to the quasi-cyclic structure, QC-LDPC codes can be encoded efficiently with shift register [4], [10]. A QC-LDPC code can be simply represented by shift values of all of its sub-matrices. This provides a compact representation of the matrix and easy construction. A circulant is a square matrix in which each row is the cyclic shift (one place to the right) of the row above it, and the first row is the cyclic shift of the last row. For such a circulant, each column is the downward cyclic shift of the column on its left, and the first column is the cyclic shift of the last column. The row and column weights of a circulant are the same, say w. For simplicity, we say that the circulant has weight w. If w = 1, then the circulant is a permutation matrix, called a circulant permutation matrix. For a circulant, the set of columns (reading top-down) is the same as the set of rows (reading from right to left). A circulant is completely characterized by its first row (or first column), which is called the generator of the circulant. A QC-LDPC code is given by the null space of an array of sparse circulants of the same size [6]. For two positive integers and with c and t with $c \leq t$, consider the following c × t array of b × b circulants over GF (2):

$$H_{qc} = \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,t} \\ A_{2,1} & A_{2,2} & \dots & A_{2,t} \\ \vdots & \vdots & \dots & \vdots \\ A_{c.1} & A_{c.2} & \dots & A_{c.t} \end{pmatrix} \quad (1)$$

The structure of a QC-LDPC code can be viewed from its parity-check matrix in circulant form, given by (1). Based on this form, every codeword v in C_{qc} (QC-LDPC code) can be divided into t sections, $v = (v_1, v_2, \dots, v_t)$, and each section v_j consists of b consecutive components of the v. For $1 \le j \le t$, the b components of the j^{th} section v_j correspond to the b columns of the j^{th} column of circulants of H_{qc}. For $0 < l < n$, let $v_j^{(l)}$ denote the vector obtained by cyclically shifting each of the b components of the j^{th} section v_j to the right l places. It is clear that $v_j^{(0)} = v_j^{(n)} = v_j$. We call $v_j^{(l)}$ the l^{th} (right) cyclic shift of v_j. Then it follows from the circulant structure of H_{qc} that the vector $v^* = (v_1^{(l)}, v_2^{(l)}, \dots, v_t^{(l)})$ is also a codeword in C_{qc}. This says that C_{qc} has sectionized cyclic structure. If the parity-check matrix H_{qc} consists of a single circulant or a single column of circulants, then C_{qc} is a cyclic code [7]. Therefore, cyclic LDPC codes [5] form a subclass of QC-LDPC codes.

Construction of Parity-Check matrix for CCSDS

The H matrices are constructed from M×M submatrices, where the submatrix size is listed in the table below.

Table 1: Values of submatrix size M for supported codes

Information block length k	Submatrix size M		
	rate ½	rate 2/3	rate 4/5
1024	512	256	128
4096	2048	1024	512
16384	8192	4096	2048

Algorithm for the calculation of generator matrix:

Step 1: Start the process by assigning the M value for the required code rate.
Step 2: Open 3 text files namely theta.txt, phi.txt, wmat.txt, theta.txt contains the θ_k values, phi.txt containing ϕ_k (m, 0-3) values and wmat.txt to write w matrix.
Step 3: Read θ_k and ϕ_k values from the files and store them in separate arrays.
Step 4: Calculate π_k values using the θ_k and ϕ_k values and store them in separate arrays.

$$\pi_k = M/4((\theta_k+(4i/M))\bmod 4)+(\phi_k[(4i/M),M]+i)\bmod(M/4) \quad (2)$$

Step 5: Construct M×M zero matrix, identity matrix and different π matrix based on k value and appending 1s at the position specified by the π_k formula (2).
Step 6: Generate the H matrix as stated in the CCSDS 131.1-O-2 doc., [8] for different code rates.
Step 7: After constructing the H matrix for desired rate, construct the W matrix which is a part of Generator matrix. W matrix is defined by the formula:

$$W = (P^{-1}Q)^T \quad (3)$$

Now from earlier generated H matrix we have to segregate P and Q matrix. These matrices are separated from H matrix as first MK columns form the Q matrix and last 3M columns form the P matrix.

Step 8: P matrix is to be inverted and this is done with the help of Gauss Jorden method. When this is achieved the inverse of P matrix is stored in an array PINV.
Step 9: The obtained PINV matrix is multiplied with Q matrix. Multiplication is normal matrix multiplication. Then the whole result is transposed by interchanging the rows and the columns.
Step10: The resultant matrix is the W matrix (3) which can be combined with identity matrix to form the G matrix.

$$G = [I_{MK} \mid W] \quad (4)$$

Step11: W matrix is written into the file Wmat.txt for further use.
Step12: Stop the process.

Note: The value of θ_k and ϕ_k are given in the CCSDS 131.1-O-2 doc [8].

Architecture of LDPC Encoder

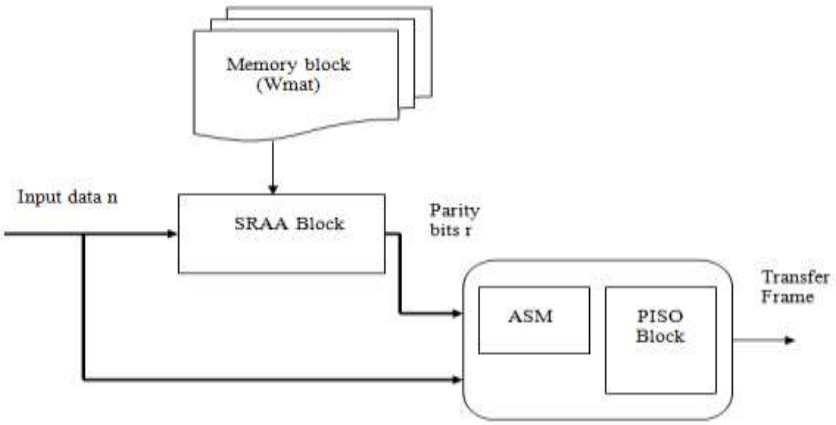


Fig.1. A schematic of the proposed encoder architecture

The overall encoder architecture is shown in Figure 1. The SRAA block consists of a shift-register-adder-accumulator circuit [4]. The memory block is mainly used to store the Wmat values that is obtain from the matlab using the algorithm described in section three. The Transfer frame block is used to combine the codeword and the Attach Sync Marker (ASM) to form a transfer frame according to the CCSDS standard [8]. This frame is then transmitted serially in the channel.

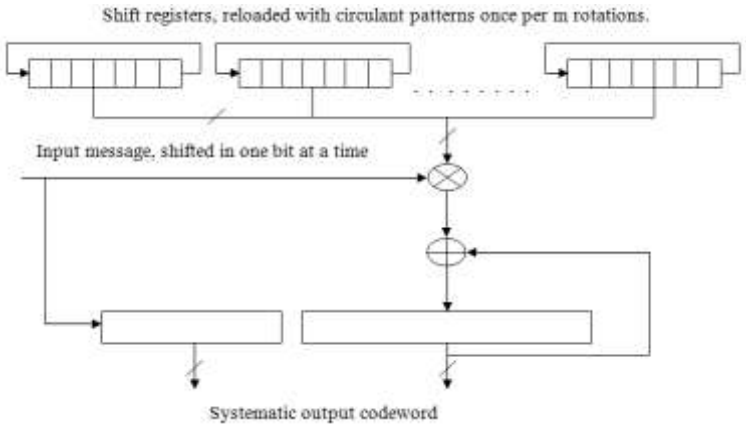
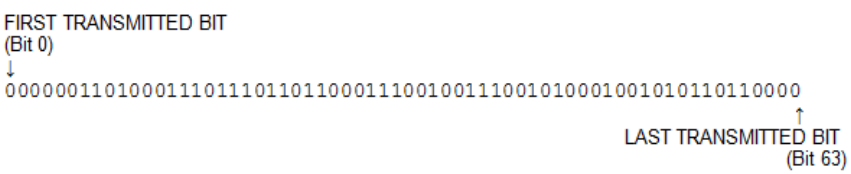


Fig.2. A Quasi-cyclic Encoder using feedback shift registers

Encoding of message m requires computing mG. Because G is block-circulant, this can be performed in an efficient bit-serial manner using linear feedback shift registers, as shown in fig. 2. Initially, the binary pattern from the first row of Wmat is placed in the shift registers. Correct endianess occurs if higher order monomials (leftmost hexadecimal digits) are placed closer to the output than the input of each shift register. Also, in terms of mG, the first bit to be encoded by the circuit represents the bit in the leftmost element of row vector m. After m bit arrivals (and cyclic shifts) then next row of Wmat is loaded. Encoding is complete after all rows of Wmat have been loaded. Each requires m clock cycles to process for a total of k clock cycles to compute the parity for one codeword. Many architectural alternatives are possible. The main benefits of the architecture in fig. 2 are conceptual simplicity and relatively high throughput (n codeword bits are computed in k clock cycles).

Codeblock synchronization is achieved by synchronization of an Attached Sync Marker associated with each LDPC Codeblock. The Attached Sync Marker (ASM) is a bit pattern, an aid to synchronization, and it precedes the LDPC Codeblock. Frame synchronizers should be set to expect a marker at a recurrence interval equal to the length of the ASM plus that of the LDPC Codeblock. All codes in the LDPC family use the 64-bit ASM.



Flowchart for the generation of parity bits of code rate 4/5;
Input length (k) = 1024 bits, Codeword length (n) = 1280bits.

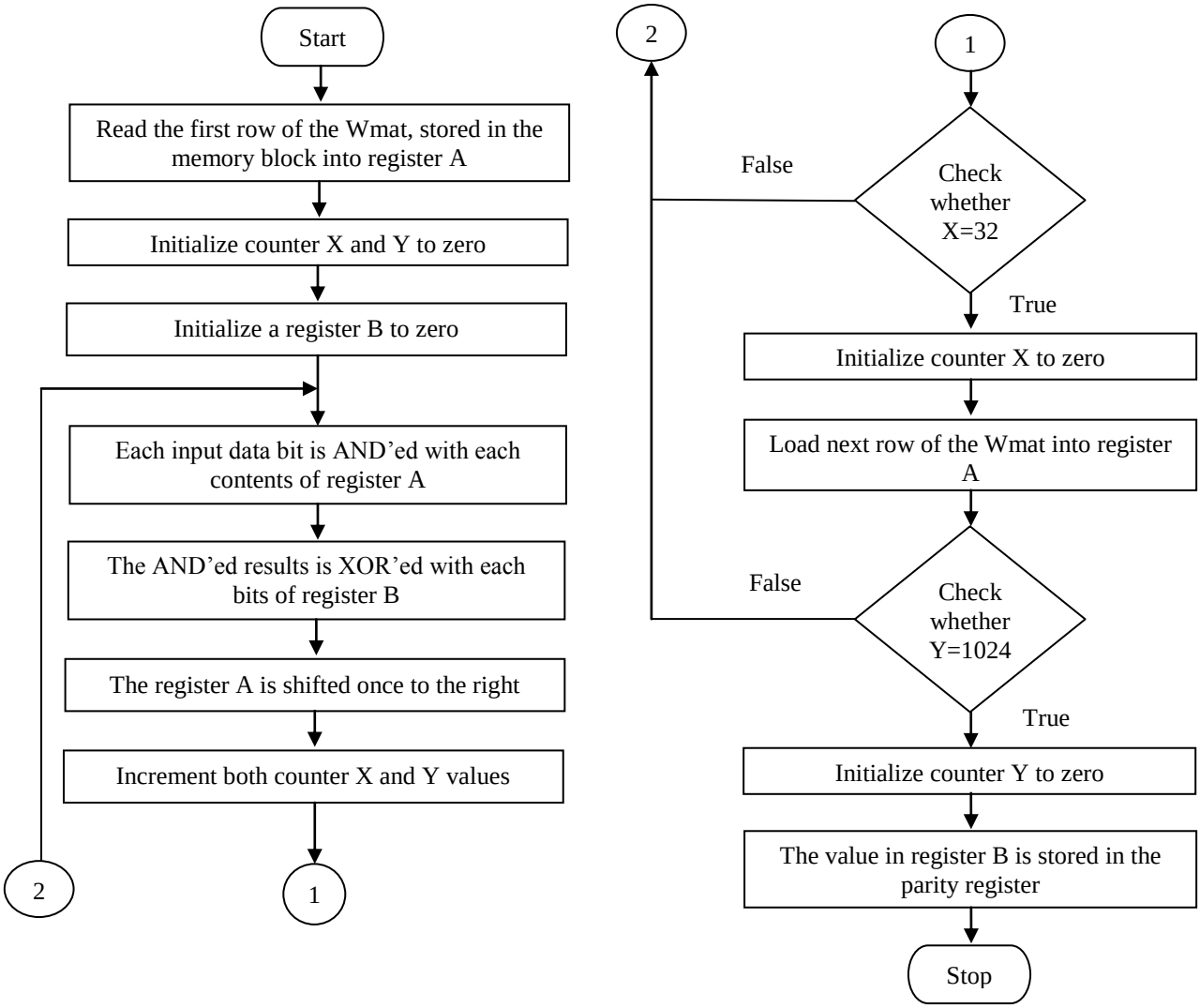


Fig.3. Flowchart for the generation of parity bits of code 4/5

Acknowledgment

The authors would like to thank the dept. of Spacecraft Checkout Group (SCG), ISRO Satellite Centre, Bangalore, India, for suggesting and supporting this research project. The authors would also like thanks to the dept. of Electronics and Communication, NMAM Institute of Technology, Nitte, for their effective guidance, support, encouragement and valuable suggestions.

Conclusion/Results

In order to evaluate the proposed architecture, a VHDL description was synthesized and tested on Altera Cyclone-2 EP2C8Q208C8 device. The code specified in the CCSDS standard 131.1-O-2 doc., [8] was used. It has maximum block length of 1024 bits. Upon testing, the highest clock frequency that would support encoding of data at code rate 4/5 was 79.59MHz. The critical path delay was dominated by the 1024-input XOR required for parity calculation. The total encoder, including all control logic occupies 4,873 logic elements and 18,432 memory bits. The synthesized report for the LDPC encoder is shown in fig.4 and also the timing analysis report is shown in fig. 5.

Architecture for Low-Density Parity-Check encoder for CCSDS standard has been presented. The algorithm described in section 3 can be used to construct the generator matrix for code rates ½, 2/3, 3/4 and 4/5, block length of 1024, 4096 and 16384. A VHDL description for code rate 4/5 with block length 1024 and codeword 1280 was synthesized and the required parity of 256 bits, for the encoder was generated. A VHDL description for other code rates and block length can also be implemented.

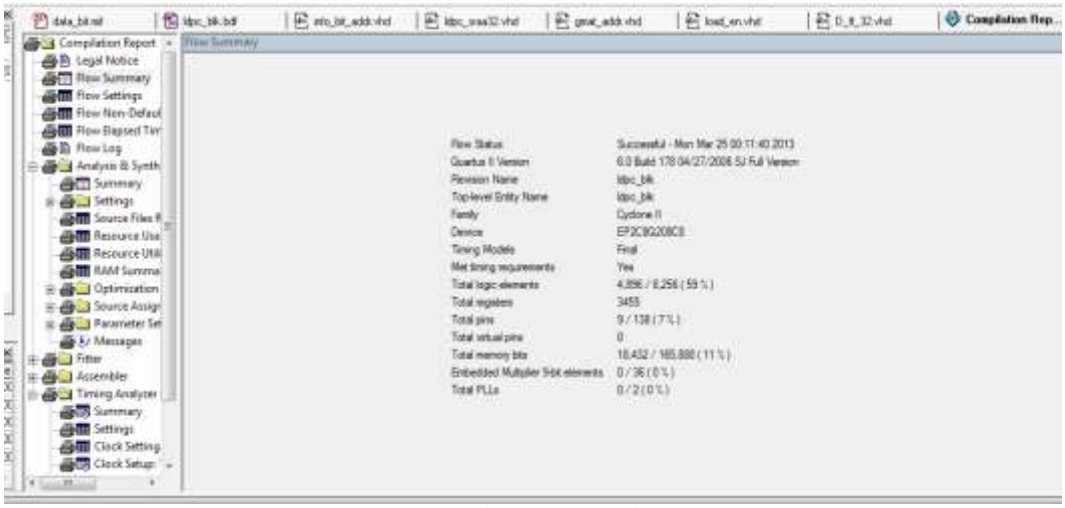


Fig.4. Synthesized report of LDPC Encoder for Cyclone-2 device

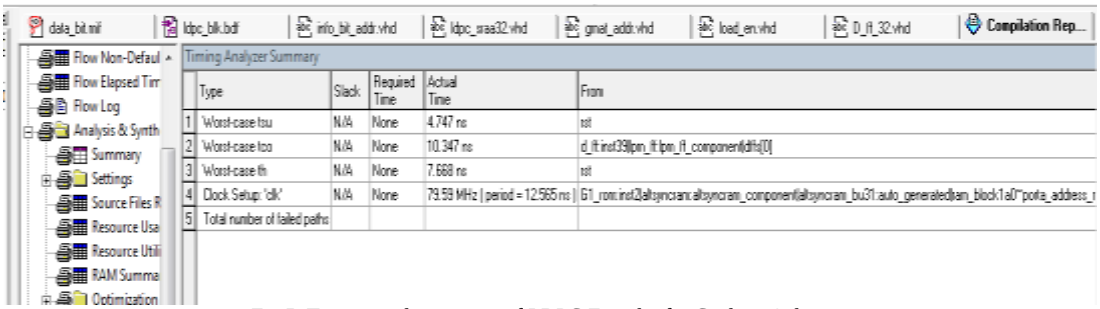


Fig.5. Timing analysis report of LDPC Encoder for Cyclone-2 device

References

[1]. R. G. Gallager, "Low density parity check codes," IRE Trans. Inf. Theory, vol. IT-8, no. 1, pp. 21–28, Jan. 1962.

[2]. D. J. C. MacKay and R. M. Neal. "Near Shannon Limit Performance of Low Density Parity Check Codes." Electro. Lett. 32 (August 1996): 1645-1646.

[3]. T. Richardson and R. Urbanke. "Design of Capacity-Approaching Low Density Parity Check Codes." IEEE Trans. Inform. Theory 47 (February 2001): 619-637.

[4]. [Z. Li, et al. "Efficient Encoding of Quasi-Cyclic Low-Density Parity-Check Codes." IEEE Transactions on Communications 54, no. 1 (January 2006): 71-81.

[5]. Y. Kou, S. Lin, and M. P. C. Fossorier. "Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results." IEEE Trans. Information Theory 47 (November 2001): 2711-2736.

[6]. L. Chen, J. Xu, I. Djurdjevic, and S. Lin, "Near-Shannon-limit quasicyclic low-density parity-check codes," IEEE Trans. Commun., vol. 52, no. 7, pp. 1038–1042, Jul. 2004.

[7]. S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.

[8]. Orange Book, Experimental specifications, Low Density Parity Check Codes For Use In Near-Earth And Deep Space Applications CCSDS 131.1-O-2. CCSDS, S ept. 2007.

[9]. D. J. C. Mackay. Information Theory, Inference, and Learning Algorithm. Cambridge Press 2003.

[10]. H. Fujeta and K. Sakaniwa. Some Classes of Quasi-Cyclic LDPC Codes: Properties and Efficient Encoding Method. IEICE Fundamentals, E88-A (12):3627-3635, 2005.

[11]. S. Lin and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2004.

[12]. W. W. Peterson and E. J. Weldon, Jr., Error Correcting Codes, 2nd ed. Cambridge, MA: MIT Press, 1972.